

Travail pratique
Physique appliquée

Simulation simple d'une galaxie en C

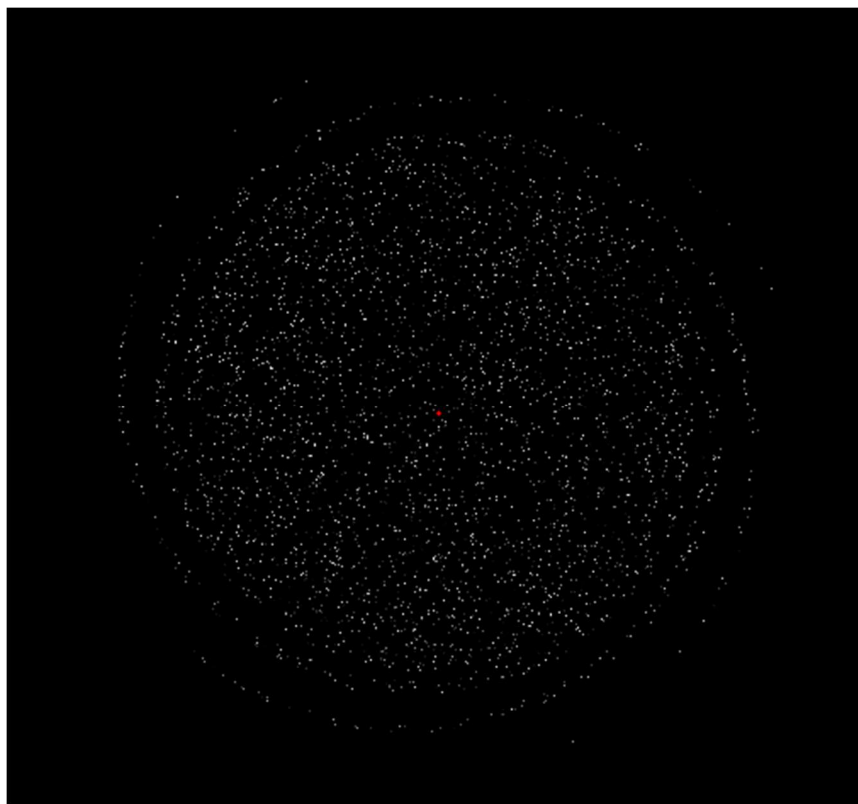
Nikola Antonijevic et Damian Boquete

Description

Ce travail pratique consistait à simuler le comportement physique d'entités astrales gravitant autour d'une autre entité bien plus massive que celles-ci. Pour se faire, toute la logique physique fut implémentée grâce au langage C et à la librairie SDL.

La simulation consiste donc à reproduire le comportement d'une galaxie. C'est-à-dire, afficher en continu le déplacement des pixels, régi par les lois de la physique, autour du pixel rouge, représentant l'entité massive (qui sera appelé « trou noir » dorénavant) mentionné au-dessus.

Voici le git du projet : <https://githopia.hesge.ch/damian.boquetec/galaxy>



Fonctionnement

Pour mener à bien cette simulation, il faut tout d'abord comprendre le comportement des astres gravitant autour d'un autre. En commençant par la force exercée sur un astre par le trou noir.

Pour bref rappel, selon la loi universelle de la gravitation, « tout corps massif exerce un champ de gravitation responsable d'une force attractive sur d'autres corps massifs ». De ce fait, il nous faut la formule nous permettant de savoir cette fameuse force exercée, (A étant le trou noir et B étant un astre quelconque) :

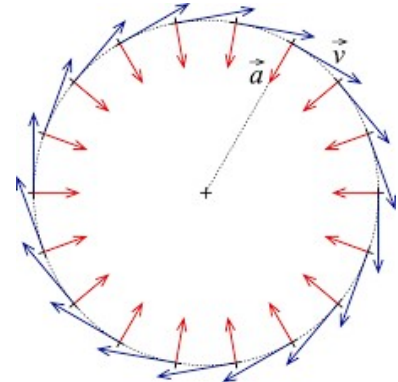
$$\vec{F}_{AB} = -G \frac{M_a M_b}{||\vec{r}_{AB}||^3} \vec{r}_{AB}$$

Avec cette force, nous pouvons déduire l'accélération de l'astre en question en divisant le vecteur \vec{F}_{AB} par sa masse M_b . Donnant ainsi :

$$\vec{a}_B = \frac{\vec{F}_{AB}}{M_B}$$

Maintenant que nous avons cette accélération pointant en direction du trou noir, il est nécessaire de calculer le mouvement circulaire uniforme de l'astre afin d'avoir un mouvement perpétuel autour de celui-ci. Cela se produit quand la vitesse de satellisation minimale et l'accélération de l'astre sont perpendiculaires.

$$v_B = \sqrt{\frac{GM}{||\vec{r}_{AB}||}}$$



La vitesse est ensuite utilisée comme scalaire du vecteur unitaire de la distance entre le trou noir et l'astre \hat{r}_{AB} . Ce vecteur unitaire \hat{r}_{AB} a une de ses composantes qui fut opposée lors de sa création afin de donner un sens de rotation.

$$\hat{r}_{AB} = \begin{pmatrix} -x / ||\vec{r}_{AB}|| \\ y / ||\vec{r}_{AB}|| \end{pmatrix}$$

$$\vec{v}_B = \hat{r}_{AB} v_B$$

Avec toutes ces formules établies, nous pouvons sans autre prédire la position future d'un astre depuis sa position actuelle. Cependant, il faut différencier la première itération (le premier mouvement après l'instant initial de la simulation) des autres. Car il faut attribuer la vitesse calculée précédemment, dans le but de lui donner le mouvement nécessaire à la rotation.

Pour la première itération nous avons donc la formule suivante :

$$pos_B(t1) = pos_B(t0) + \vec{v}_B(t0)\Delta t + \vec{a}_B(t0)\Delta t^2$$

Une fois la première position suivant l'initiale calculée, la mise à jour des suivantes s'effectue donc ainsi :

$$pos_B(ti + 1) = 2pos_B(ti) - pos_B(ti - 1) + \vec{a}_B(t0)\Delta t^2$$

Maintenant que la partie physique est établie, il nous faut l'imaginer et le structurer en C. Nous avons commencé par initialiser la galaxie avec la fonction *setupGalaxy()*, créant le trou noir et un tableau d'astres.

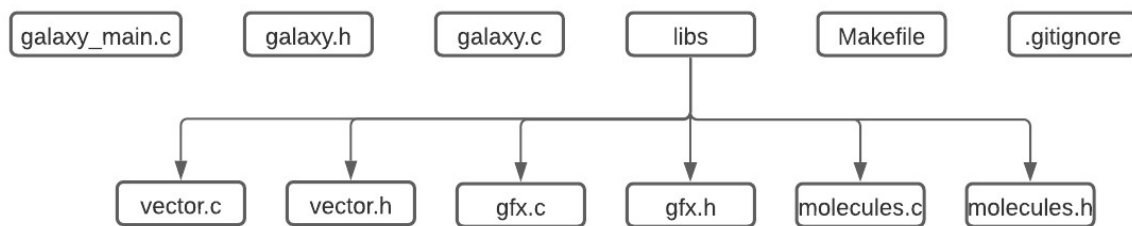
S'ensuit la création aléatoire des positions des astres avec *randomizeStarsPositions()*, itérant dans l'entièreté du tableau d'astres et attribuant aux composantes de leur position une valeur aléatoire dans une intervalle de -150px à 150px à partir du milieu de l'écran SDL.

La vitesse d'un astre est calculée et attribuée par *computeVelocity()*, le reste des calculs s'effectuent dans *computeNewPosition()*, qui s'occupe notamment de la force résultante, l'accélération, l'appel à la fonction *computeVelocity()* et, comme son nom l'indique, de déduire la nouvelle position de l'astre.

Finalement, les calculs, l'affichage des astres et du trou noir sur un écran SDL s'exécute depuis la fonction *displayGalaxy()*. Une petite fonction nommée *formatBlackhole()* s'occupe de rendre le trou noir plus visible en l'agrandissant.

Structure du projet

Nous avons utilisé une structure de fichier assez classique pour ce projet. Le dossier principal contient le fichier *galaxy_main.c*, contenant la fonction *main()* de notre application. Ensuite il y a *galaxy.h* et *galaxy.c* qui contiennent les fonctions qui vont permettre d'initialiser la galaxie, de créer des étoiles à des positions aléatoires ainsi que certains calculs comme la fonction qui permet de calculer la rotation d'une étoile ou son accélération. Finalement, il reste la fonction qui permet l'affichage de notre projet.



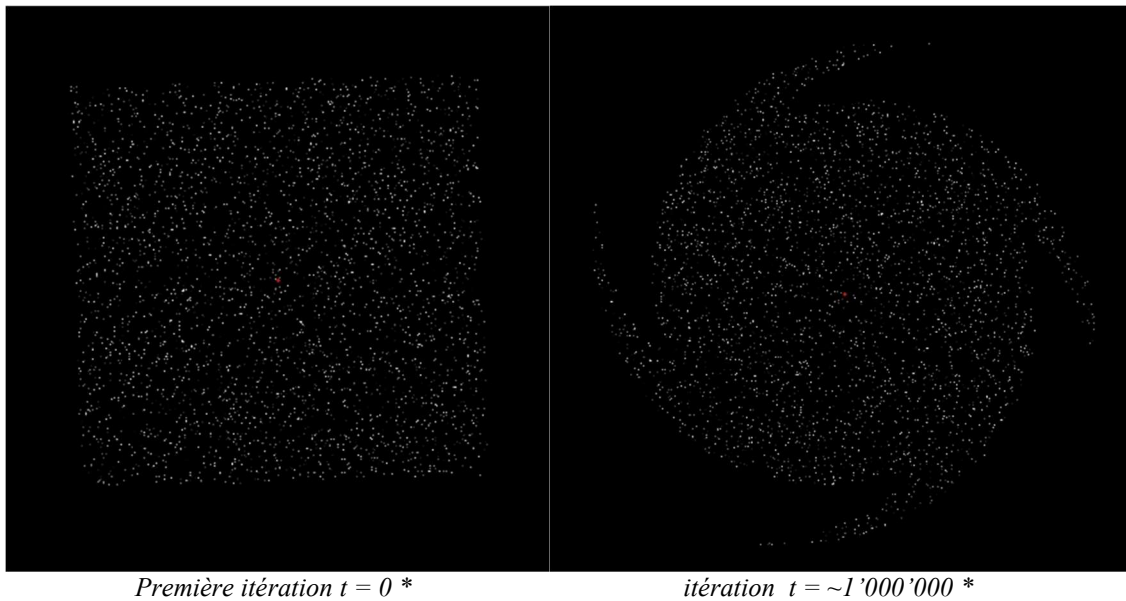
Arborescence du projet

Le projet étant développé en C, un *Makefile* était nécessaire afin de gagner du temps à la compilation du projet. Il y a également un fichier *.gitignore* qui nous permet d'utiliser git sans avoir les fichiers poubelles dans les commit.

Le dossier *libs* contient les différentes librairies que nous avons utilisées pour développer le projet. Premièrement, nous avons utilisé *vector.h* et *vector.c* pour toutes les opérations liées aux vecteurs. Afin d'afficher le résultat, il nous fallait utiliser les fichiers *gfx.h* et *gfx.c* qui font partie de la librairie SDL.

Pour finir, nous nous sommes servis de *molecules.h* et *molecules.c* qui nous ont permis de créer une structure pour les molécules.

Figures et résultats



```

T = 1000

Masse : 7.000000
Ancienne position[x][y] : [-1.000000][-1.000000]
Position[x][y] : [261.000000][304.000000]
Vitesse : [1.000000][0.000000]
Accélération[x][y] : [0.000000e+00.10][0.000000e+00.10]
FRes[x][y] : [0.000000e+00.10][0.000000e+00.10]

T = 2000

Masse : 7.000000
Ancienne position[x][y] : [261.000000][304.000000]
Position[x][y] : [261.017664][304.130762]
Vitesse : [0.000017][0.000131]
Accélération[x][y] : [4.503727e-10.10][-5.925956e-11.10]
FRes[x][y] : [3.152609e-09.10][3.152609e-09.10]

```

Évolution des valeurs d'une molécule entre $t = 1'000$ et $t = 2'000$ *

* $\Delta t = 1'000$

Sources

Image du MCU (page 3)

https://www.google.com/url?sa=i&url=http%3A%2F%2Fphysique-appliquee.net%2Fbanque_images%2Fpicture.php%3F%2F3415&psig=AOvVaw0sjhSNcma3mXv03Vy2eKU-&ust=1609688346962000&source=images&cd=vfe&ved=0CAIQjRxqFwoTCNilmsXK_e0CFQAAAAAdAAAAABAD